IFW
AF

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. __10002673-1__

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Shinobu Togasaki	Confirmation No.: 2701
Application No.:	10/020,355	Examiner: Anya, Charles E.
Filing Date:	October 29, 2001	Group Art Unit: 2194

Title: **APPARATUS AND METHOD FOR ROUTING A TRANSACTION TO A SERVER**

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on __October 12, 2005__ .

The fee for filing this Appeal Brief is (37 CFR 1.17(c)) $500.00.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐ 1st Month $120	☐ 2nd Month $450	☐ 3rd Month $1020	☐ 4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of ___$ 500___ . At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

☒ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit: December 12, 2005

OR

☐ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.

Date of facsimile:

Typed Name: Chasity C. Rossum
Signature:

Respectfully submitted,

Shinobu Togasaki

By

Gregory W. Osterloth

Attorney/Agent for Applicant(s)

Reg No. :	36,232
Date :	December 12, 2005
Telephone :	(303) 291-3204

Rev 10/05 (AplBrief)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appl. No.   :   10/020,355   Confirmation No. 2701
Appellant   :   Shinobu Togasaki
Filed   :   October 29, 2001
TC/A.U.   :   2194
Examiner   :   Anya, Charles E.

Docket No.   :   10002673-1

Board of Patent Appeals and Interferences
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Table of Contents

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Appl. No.	:	10/020,355	Confirmation No. 2701
Appellant	:	Shinobu Togasaki	
Filed	:	October 29, 2001	
TC/A.U.	:	2194	
Examiner	:	Anya, Charles E.	

Docket No.   :   10002673-1

Board of Patent Appeals and Interferences
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Dear Sir:

This Appeal Brief is submitted in response to the Examiner's Final Office Action dated June 13, 2005.

Appellants filed a Notice of Appeal on October 12, 2005.

Real Party in Interest

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, Texas 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, California. The general or managing partner of HPDC is HPQ Holdings, LLC.

Related Appeals and Interferences

There are no related appeals and/or interferences.

Status of Claims

Claims 1-33 remain in the Application, all of which stand rejected. A copy of the claims is provided in the attached Claims Appendix.

Status of Amendments

No amendments have been filed since the Final Office Action.

Summary of Claimed Subject Matter

Claim 1: A method for routing a transaction (p. 5, lines 29-30; FIG. 1, 100) to a front-end server (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125), comprising:

identifying at least one attribute-based category (p. 5, line 28-29; FIG. 4, 410) for said transaction (p. 5, lines 29-30; FIG. 1, 100);

identifying at least one of a plurality of front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) to process said transaction (p. 5, lines 29-30; FIG. 1, 100) based at least in part on said identified attribute-based category (p. 5, line 28-29; FIG. 4, 410) of said transaction (p. 5, lines 29-30; FIG. 1, 100) and at least in part on said front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) being assigned to execute transactions corresponding to said attribute-based category (p. 6, lines 3-9; p. 9, line 32 - p. 10, line 8); and

routing said transaction (p. 5, lines 29-30; FIG. 1, 100) to one of said at least one identified front-end servers (p. 4, line 33).

Claim 10: An apparatus for routing a transaction (p. 5, lines 29-30; FIG. 1, 100) to a front-end server (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125), comprising:

computer readable storage media (p. 5, lines 24-27);

computer readable program code stored on said computer readable storage media (p. 5, lines 24-27), comprising:

a)   program code for identifying at least one attribute-based category (p. 5, line 28-29; FIG. 4, 410) for said transaction (p. 5, lines 29-30; FIG. 1, 100);

b)   program code for identifying at least one of a plurality of front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) to process said transaction (p. 5, lines 29-30; FIG. 1, 100) based at least in part on said identified attribute-based category (p. 5, line 28-29; FIG. 4, 410) of said transaction (p. 5, lines 29-30; FIG. 1, 100) and at least in part on said front-

end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) being assigned to execute transactions corresponding to said attribute-based category(p. 5, line 28-29; FIG. 4, 410); and

c)   program code for routing said transaction (p. 5, lines 29-30; FIG. 1, 100) to one of said at least one identified front-end server.

Claim 21: An apparatus for routing a transaction (p. 5, lines 29-30; FIG. 1, 100) to a server (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125), comprising:

means for identifying at least one attribute-based category (p. 5, line 28-29; FIG. 4, 410) for said transaction (p. 5, lines 29-30; FIG. 1, 100);

means for identifying at least one of a plurality of servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) to process said transaction (p. 5, lines 29-30; FIG. 1, 100) based at least in part on said identified attribute-based category (p. 5, line 28-29; FIG. 4, 410) of said transaction (p. 5, lines 29-30; FIG. 1, 100) and at least in part on said servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) being assigned to execute transactions corresponding to said attribute-based category (p. 5, line 28-29; FIG. 4, 410); and

means for routing said transaction (p. 5, lines 29-30; FIG. 1, 100) to one of said at least one identified servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125).

Claim 33: A method for routing a transaction (p. 5, lines 29-30; FIG. 1, 100) to a front-end server, comprising:

maintaining a table (p. 14, lines 13-15; FIG. 3, 370) at a workload manager (p. 6, lines 17-18; FIG. 1, 110-112), the table (p. 14, lines 13-15; FIG. 3, 370) comprising indications of which attribute-based categories (p. 5, line 28-29; FIG. 4, 410) of transactions (p. 5, lines 29-30; FIG. 1, 100) are assigned to which of a plurality of front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125);

periodically updating the table in response to broadcasts received from said front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125);

upon receiving said transaction (p. 5, lines 29-30; FIG. 1, 100) at the workload manager (p. 6, lines 17-18; FIG. 1, 110-112),

identifying at least one attribute-based category (p. 5, line 28-29; FIG. 4, 410) for the transaction (p. 5, lines 29-30; FIG. 1, 100);

identifying at least one of the plurality of front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125) to process the transaction (p. 5, lines 29-30; FIG. 1, 100) based at least in part on said identified attribute-based category (p. 5, line 28-29; FIG. 4, 410) of said transaction (p. 5, lines 29-30; FIG. 1, 100) and at least in part on whether said table comprises an indication that said identified attribute-based category (p. 5, line 28-29; FIG. 4, 410) is assigned to one of said front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125); and

routing said transaction (p. 5, lines 29-30; FIG. 1, 100) to one of said at least one identified front-end servers (p. 4, line 33-p. 5, line 1; FIGs. 2-3, 121-125).

Grounds of Rejection to be Reviewed on Appeal

1.   Whether claims 1-8, 10-18 and 20-29 should be rejected under 35 USC 102(b) as being anticipated by Kanai et al. (U.S. Pat. No. 5,864,679 of Kanai et al.).

2.   Whether claims 9 and 19 should be rejected under 35 USC 103(a) as being unpatentable over Kanai et al. (U.S. Pat. No. 5,864,679 of Kanai et al.) in view of Cross et al. (U.S. Pat. No. 6,681,244).

3.   Whether claims 30-33 should be rejected under 35 USC 103(a) as being unpatentable over Kanai et al. (U.S. Pat. No. 5,864,679 of Kanai et al.) in view of Shapiro et al. (U.S. Pub. No. 2002/0161917).

Arguments

1.   Whether claims 1-8, 10-18 and 20-29 should be rejected under 35 USC 102(b) as being anticipated by Kanai et al. (U.S. Pat. No. 5,864,679 of Kanai et al.; hereinafter "Kanai").

a. Claims 1-6, 10-12, 15-18 and 20-27:

Appellant's claim 1 recites:

1.   A method for routing a transaction to a front-end server, comprising:
        identifying at least one attribute-based category for said transaction;
        identifying at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category; and
        routing said transaction to one of said at least one identified front-end servers.

With respect to Appellant's claim 1, the Examiner asserts that "routing a transaction to a front-end server" is generally taught by Kanai in FIG. 3; col. 10, lines 40-67; FIG. 4; col. 13, lines 61-67; FIG. 5; and col. 14, lines 40-52. The Examiner also asserts that "identifying at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category" is taught by Kanai in col. 15, lines 56-62. Appellant disagrees.

In col. 15, line 17 – col. 16, line 11, Kanai teaches the use of a data arrangement table 4B and routing table 4C by a transaction routing unit 4 (*see also*, FIGS. 7-9). Appellant notes, however, that neither of these tables (i.e., table 4B or 4C) maintains assignments between front-end servers and attribute-based categories of transactions. Rather, Kanai uses a transaction type to index

table 4C (FIG. 9) and determine which of a number of transaction arguments should be used to specifically identify the data required by the transaction. Kanai then uses these transaction arguments to index table 4B (FIG. 8) and determine which of a number of transaction processors is specifically assigned to access the data required by the transaction.

In contrast to what Kanai discloses, the invention of Appellant's claim 1 routes a transaction based on whether an attribute-based category that is associated with the transaction has been assigned to a particular front-end server. Thus, instead of determining whether the data required by a transaction "is" at a certain processor (as Kanai teaches), the invention of Appellant's claim 1 projects where data required by a transaction "may be" based on a server's association with a particular attribute-based "category". Although the invention of Appellant's claim 1 may at times be less accurate than Kanai's method (which, according to Kanai, is a "deterministic algorithm"; see, col. 15, lines 65-67), Appellant's method can often provide close to the same accuracy, but with faster routing and less overhead.

In some ways, the invention of Appellant's claim 1 is more akin to Kanai's "probabilistic algorithm" (see, col. 11, line 64 – col. 12, line 20). However, in contrast to Kanai's maintenance of the processing history and processing cost for each of a number of routed transactions, the invention of Appellant's claim 1 routes transactions based on assignments of attribute-based transaction "categories" to particular front-end servers.

In response to the above arguments, the Examiner indicated in his Final Office Action that, "the invention as claimed requires the selection of a front-end server based. . .in part on the front-end server *corresponding to* the attribute-based category." See, 6/13/2005 Final Office Action, sec. 47, p. 12 (emphasis added). Appellant respectfully disagrees. What Appellant's claim 1 recites is, "identifying at least one of a plurality of front-end servers to process said transaction based. . .at least in part on said front-end servers *being assigned to execute transactions corresponding to* said attribute-based category" (emphasis added). Appellant draws the Examiner's attention to this difference in

language because Kanai does not teach that servers are assigned to execute transactions corresponding to attribute-based categories of transactions. Rather, and as discussed above, Kanai teaches 1) a "deterministic algorithm" for routing transactions, wherein transactions are routed based on where the data they need "is", and 2) a "probabilistic algorithm" for routing transactions, wherein transactions are routed based on specific transaction processing histories and processing costs. In either case, Kanai does not teach that a server is *assigned* to execute transactions corresponding to a particular *attribute-based category*.

Although the historical routing information maintained by Kanai's "probabilistic algorithm" does include "feature parameters" for routed transactions, it is noted that both feature parameters, and the servers to which transactions are routed, are associated with specific historical transactions. Thus, instead of just routing a transaction based on 1) "identifying at least one attribute-based category for said transaction", and then 2) "identifying at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category" (as set forth in Appellant's claim 1), Kanai's routing algorithm must 1) identify feature parameters for a transaction, 2) identify historical transactions associated with the identified feature parameters, and do so for each of a plurality of transaction processors, and 3) weigh i) how close the feature parameters of the identified historical transactions are to those of the current transaction that needs to be routed, against ii) the cost of routing the current transaction to each of the transaction processors associated with the identified historical transactions. *See*, for example, Kanai's teachings at col. 20, line 62 – col. 21, line 44. The complexity of Kanai's routing algorithm stems, at least in part, from the fact that Kanai does not teach the *assignment* of servers to execute transactions corresponding to different attribute-based categories.

Appellant's claim 1 is believed to be allowable for at least the above reasons. Appellant's claims 2-6 and 23-27 are believed to be allowable at least

for the reason that they depend from Appellant's claim 1. Appellant's claims 10-12, 15-18 and 20-22 are believed to be allowable at least for reasons similar to why Appellant's claim 1 is believed to be allowable.

b. Claim 7:

Appellant's claim 7 recites:

7.   A method as in claim 1, further comprising determining when said identified attribute-based category is new and assigning said new attribute-based category to at least one of said plurality of front-end servers.

With respect to Appellant's claim 7, the Examiner asserts that Kanai teaches "determining when said identified attribute-based category is new and assigning said new attribute-based category to at least one of said plurality of front-end servers" in col. 15, lines 1-25. Appellant disagrees. What Kanai teaches is how to *route* a "newly arrived transaction", and not how to *assign* a new attribute-based category to a front-end server.

Appellant's claim 7 is believed to be allowable for at least the above reasons, and because it depends from Appellant's claim 1 (*see*, sec. 1.a. of these Arguments, *supra*).

c. Claim 8:

Appellant's claim 8 recites:

8.   A method as in claim 7, further comprising notifying a workload manager of said at least one front-end server assigned to said new attribute-based category.

With respect to Appellant's claim 8, the Examiner asserts that Kanai teaches "notifying a workload manager of said at least one front-end server assigned to said new attribute-based category" in col. 15, lines 17-32. Appellant disagrees. What Kanai teaches is that upon a determination of a "new *data arrangement*" amongst the transaction processors, this arrangement may be provided to the transaction routing unit. This is not the same as notifying a workload manager of a new *attribute-based category being assigned to a front-end server.*

Appellant's claim 8 is believed to be allowable for at least the above reasons, and because it depends from Appellant's claim 1 (*see*, sec. 1.a. of these Arguments, *supra*).

d. Claim 13:

Appellant's claim 13 recites:

13.   An apparatus as in claim 10, wherein said attribute-based category is based on at least one "perceived" attribute of said transaction.

With respect to Appellant's claim 13, the Examiner asserts that, in col. 15, lines 26-62, and in FIGS. 9 & 10, Kanai teaches an attribute-based category being based on at least one "perceived" attribute of a transaction. Appellant disagrees. Kanai says nothing about real versus perceived attributes of a transaction. Although Kanai's method can rely on transaction arguments such as a "Teller-ID" or a "Branch-ID", these arguments are used to determine which of a number of transaction processors is assigned to the data required by a transaction. These arguments (Teller-ID and Branch-ID) are not assigned to a particular server to which all transactions having the same attribute-based category are routed.

Appellant's claim 13 is believed to be allowable for at least the above reasons, and for reasons similar to why Appellant's claim 1 is believed to be allowable (*see*, sec. 1.a. of these Arguments, *supra*).

e. Claim 14:

Appellant's claim 14 recites:

14.   An apparatus as in claim 10, further comprising a user table for assigning said at least one attribute-based category to said transaction.

With respect to Appellant's claim 14, the Examiner asserts that Kanai teaches "a user table for assigning said at least one attribute-based category to said transaction" in col. 15, lines 45-62. Appellant disagrees. What Kanai discusses in this paragraph is how to look up transaction arguments in a routing table. The user table recited in Appellant's claim 14 is for assigning the attribute-based category, which has to be done before the assigned attribute-based category can be used to look something up. Providing a user table for assigning an attribute-based category to a transaction implies that the transaction does not automatically carry this information, which Appellant believes to be novel when combined with the code for routing a transaction described in his claim 10.

Appellant's claim 14 is believed to be allowable for at least the above reasons, and for reasons similar to why Appellant's claim 1 is believed to be allowable (*see*, sec. 1.a. of these Arguments, *supra*).

f. Claims 28 and 29:

Appellant's claims 28 and 29 recite:

28.   A method as in claim 27, further comprising:

if the identified front-end server establishes an association between itself and an attribute-based category, broadcasting this association to a plurality of workload managers that can route transactions to the identified front-end server.

29. A method as in claim 28, further comprising:
upon a workload manager's receipt of said broadcast association, the workload manager updating its own table of assignments between attribute-based categories and front-end servers.

With respect to claims 28 and 29, the Examiner asserts that Kanai teaches the broadcast of server assignments to a plurality of workload managers, in col. 25, lines 39-46. Appellant disagrees. Given that Kanai does not teach the assignment of attribute-based transaction categories to servers, Kanai cannot teach the broadcast of same. Furthermore, the "registered" transaction and transaction source processor referred to by Kanai are not server assignments. Nor do they appear to be "broadcast" to *a plurality of* workload managers.

Appellant's claim 28 and 29 are believed to be allowable for at least the above reasons, and because they depend from Appellant's claim 1 (*see*, sec. 1.a. of these Arguments, *supra*).

2. **Whether claims 9 and 19 should be rejected under 35 USC 103(a) as being unpatentable over Kanai in view of Cross et al. (U.S. Pat. No. 6,681,244; hereinafter "Cross").**

Appellant's claim 9 recites:

9. A method as in claim 1, further comprising:
determining a status of an attribute-based category; and
deallocating said attribute-based category from said front-end server to which it is assigned when said status is inactive.

With respect to Appellant's claim 9, the Examiner asserts that Cross teaches "determining a status of an attribute-based category; and deallocating said attribute-based category from said front-end server to which it is assigned when said status is inactive" in col. 6, lines 15-27. Appellant disagrees. What

- 17 -

Cross teaches is a switch's removal of a client machine's address from its network table if the switch does not detect a data packet from the client within a predetermined time interval. Cross' switch is not a front-end server. Nor does Cross teach or suggest how its switch might be related to the transaction routing method taught by Kanai. Claims 9 and 19 are therefore believed to be allowable in that a combination of Kanai's and Cross' teachings does not yield the inventions of these claims. These claims are also believed to be allowable for the reasons that their parent claims are believed to be allowable. *See*, sec. 1.a. of these Arguments.

3.   Whether claims 30-33 should be rejected under 35 USC 103(a) as being unpatentable over Kanai in view of Shapiro et al. (U.S. Pub. No. 2002/0161917; hereinafter "Shapiro").

Appellant's claim 30 recites:

30.   A method as in claim 1, further comprising:
one or more of said front-end servers,
            maintaining its own table of attribute-based categories for transactions that it has processed;
            for each attribute-based category in its table, maintaining an indication of when a transaction corresponding to the attribute-based category was last processed by the front-end server; and
            after a predetermined time of not processing a transaction corresponding to an attribute-based category in its table, broadcasting an indication of this event to a plurality of workload managers that can route transactions to the front-end server.

With respect to claim 30, the Examiner equates a front-end server's maintenance of "its own table of attribute-based categories for transactions that [a sever] has processed" with Kanai's History Information Management Unit 107 and History Information Memory Unit 108. Appellant respectfully disagrees. Referring to Kanai's FIG. 11, it is clear that the Units 107 and 108 are part of Kanai's Transaction Routing Unit 101, and not part of any Transaction Processor 110-1 to 110-m (or front-end server).

Also with respect to claim 30, the Examiner asserts that Shapiro teaches, "after a predetermined time of not processing a transaction corresponding to an attribute-based category in its table, broadcasting an indication of this event to a plurality of workload managers that can route transactions to the front-end server (". . .poor goodness. . ." page 6 paragraphs 0070/0071)." *See*, 6/13/2005 Final Office Action, sec. 39, p. 9. Appellant respectfully disagrees. The cited paragraphs (0070/0071) of Shapiro say nothing about 1) attribute-based transaction categories, 2) the assignment of attribute-based transaction categories to front-end server, or 3) the broadcast of such assignments *from front-end servers*.

In light of the deficiencies of both Kanai's and Shapiro's teachings, Appellant believes claim 30 is allowable. Each of Appellant's claims 31-33 is believed to be allowable at least for the above reasons, or for reasons similar to the above reasons.

4. Conclusion

In summary, the art of record does not teach nor suggest the subject matter of Appellants' claims 1-33. These claims are therefore believed to be allowable, and accordingly, Appellants respectfully request the issuance of a Notice of Allowance.

Respectfully submitted,
DAHL & OSTERLOTH, L.L.P.

By
Gregory W. Osterloth
Reg. No. 36, 232
Tel: (303) 291-3200

Claims Appendix

Listing of Claims:

Claim 1: A method for routing a transaction to a front-end server, comprising:

identifying at least one attribute-based category for said transaction;

identifying at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category; and

routing said transaction to one of said at least one identified front-end servers.

Claim 2: A method as in claim 1, further comprising assigning said at least one attribute-based category to said transaction.

Claim 3: A method as in claim 2, wherein assigning said at least one attribute-based category to said transaction comprises associating a tag with said transaction.

Claim 4: A method as in claim 1, wherein identifying said at least one front-end server comprises comparing said attribute-based category for said transaction to assigned attribute-based categories for said plurality of front-end servers.

Claim 5: A method as in claim 1, further comprising determining whether said at least one front-end server is available for processing said transaction.

Claim 6: A method as in claim 1, further comprising rerouting said transaction to another of said plurality of front-end servers when said identified server refuses said transaction.

Claim 7: A method as in claim 1, further comprising determining when said identified attribute-based category is new and assigning said new attribute-based category to at least one of said plurality of front-end servers.

Claim 8: A method as in claim 7, further comprising notifying a workload manager of said at least one front-end server assigned to said new attribute-based category.

Claim 9: A method as in claim 1, further comprising:

determining a status of an attribute-based category; and

deallocating said attribute-based category from said front-end server to which it is assigned when said status is inactive.

Claim 10: An apparatus for routing a transaction to a front-end server, comprising:

computer readable storage media;

computer readable program code stored on said computer readable storage media, comprising:

d)   program code for identifying at least one attribute-based category for said transaction;

e)   program code for identifying at least one of a plurality of front-end servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said front-end servers being assigned to execute transactions corresponding to said attribute-based category; and

f)   program code for routing said transaction to one of said at least one identified front-end server.

Claim 11: An apparatus as in claim 10, further comprising program code for assigning said at least one attribute-based category to said transaction.

Claim 12: An apparatus as in claim 10, wherein said attribute-based category is based on at least one "real" attribute of said transaction.

Claim 13: An apparatus as in claim 10, wherein said attribute-based category is based on at least one "perceived" attribute of said transaction.

Claim 14: An apparatus as in claim 10, further comprising a user table for assigning said at least one attribute-based category to said transaction.

Claim 15: An apparatus as in claim 10, further comprising:

program code for determining whether said at least one identified server is available for processing said transaction; and

program code for rerouting said transaction to another of said plurality of servers when at least one identified server is unavailable for processing said transaction.

Claim 16: An apparatus as in claim 10, further comprising program code for assigning a number of attribute-based categories to each of said plurality of front-end servers, wherein said program code for routing said transaction to one of said identified front-end servers routes said transaction according to said assigned attribute-based categories.

Claim 17: An apparatus as in claim 16, wherein said program code for assigning at least one attribute-based category to each of said plurality of servers bases the assignment at least in part on an affinity of transaction attributes.

Claim 18: An apparatus as in claim 16, further comprising a workload manager table for recording said assigned attribute-based categories.

Claim 19: An apparatus as in claim 16, further comprising:

program code for determining a status for each of said assigned attribute-based categories; and

program code for deallocating said assigned attribute-based categories when said status thereof is inactive.

Claim 20: An apparatus as in claim 10, further comprising program code for determining when said identified attribute-based category is new and assigning said new attribute-based category to at least one of said plurality of front-end servers.

Claim 21: An apparatus for routing a transaction to a server, comprising:

means for identifying at least one attribute-based category for said transaction;

means for identifying at least one of a plurality of servers to process said transaction based at least in part on said identified attribute-based category of said transaction and at least in part on said servers being assigned to execute transactions corresponding to said attribute-based category; and

means for routing said transaction to one of said at least one identified servers.

Claim 22: An apparatus as in claim 21, further comprising:

means for identifying each of said plurality of servers; and

means for assigning at least one attribute-based category to each of said plurality of servers.

Claim 23: A method as in claim 1, wherein identifying said at least one attribute-based category for said transaction comprises identifying a "perceived" attribute of said transaction.

Claim 24: A method as in claim 23, wherein the identified "perceived" attribute is the computer originating the transaction.

Claim 25: A method as in claim 23, wherein the identified "perceived" attribute is the user originating the transaction.

Claim 26: A method as in claim 23, wherein the identified "perceived" attribute is a class of users from which the transaction originates.

Claim 27: A method as in claim 1, wherein said identifying and routing actions are performed by a workload manager, the method further comprising:

determining, at an identified front-end server, whether the attribute-based category associated with said received transaction is assigned to the identified front-end server, and if it is not, establishing an association between i) the attribute-based category of the received transaction and ii) the identified front-end server.

Claim 28: A method as in claim 27, further comprising:

if the identified front-end server establishes an association between itself and an attribute-based category, broadcasting this association to a plurality of workload managers that can route transactions to the identified front-end server.

Claim 29: A method as in claim 28, further comprising:

upon a workload manager's receipt of said broadcast association, the workload manager updating its own table of assignments between attribute-based categories and front-end servers.

Claim 30: A method as in claim 1, further comprising:

one or more of said front-end servers,

maintaining its own table of attribute-based categories for transactions that it has processed;

for each attribute-based category in its table, maintaining an

indication of when a transaction corresponding to the attribute-based

category was last processed by the front-end server; and

after a predetermined time of not processing a transaction

corresponding to an attribute-based category in its table, broadcasting an

indication of this event to a plurality of workload managers that can route

transactions to the front-end server.

Claim 31: A method as in claim 30, further comprising:

upon a workload manager's receipt of said broadcast indication, the

workload manager updating its own table of assignments between attribute-

based categories and front-end servers.

Claim 32: An apparatus as in claim 10, further comprising program code to

update, in response to broadcast indications from said front-end servers, a table

of which attribute-based categories are assigned to which front-end servers, said

table being maintained by and for a particular workload manager.

Claim 33: A method for routing a transaction to a front-end server, comprising:

maintaining a table at a workload manager, the table comprising

indications of which attribute-based categories of transactions are assigned to

which of a plurality of front-end servers;

periodically updating the table in response to broadcasts received from

said front-end servers;

upon receiving said transaction at the workload manager,

identifying at least one attribute-based category for the transaction;

identifying at least one of the plurality of front-end servers to

process the transaction based at least in part on said identified attribute-

based category of said transaction and at least in part on whether said

table comprises an indication that said identified attribute-based category

is assigned to one of said front-end servers; and

routing said transaction to one of said at least one identified front-end servers.

Evidence Appendix

No extrinsic evidence has been entered and relied upon in this appeal.

Related Proceedings Appendix

There are no related proceedings in any court or before the Board.

Carlsbad Caverns National Park, NM

IP Administration
Legal Department, MS 35
HEWLETT-PACKARD COMPANY
P.O. Box 272400
Fort Collins, Colorado 80527-2400

© USPS 2002 ♻ recycled

RECEIVED IN RE: **CONFIRMATION NO.:** 2701
APPLICATION NO.:   10/020,355
APPLICANT(S):   Shinobu Togasaki
FILED: October 29, 2001
FOR: APPARATUS AND METHOD FOR ROUTING A TRANSACTION TO A
                         SERVER
ITEMS ENCLOSED:
- Transmittal of Appeal Brief
- Appeal Brief
- Postcards for return by the United States Patent and Trademark Office.

Certificate of Mailing for First Class Mail
Mailing Date: December 12, 2005
Client Matter No.:   HP 10002673-1
Attorney Initials: GWO/CCR

Please place the official stamp of the Patent Office on this card and return it to us for our files to constitute
an acknowledgment by the Patent Office of receipt of the above-identified papers on the date stamped.